Appendix I

T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet

Dan Venolia

Apple Computer, Inc.
1 Infinite Loop
Cupertino CA 95014 USA
(408) 974-4121
E-mail: danv@apple.com

Forrest Neiberg

2843 East Grand River #157
East Lansing MI 48823 USA
E-mail: fneiberg@mcimail.com

ABSTRACT
An interface for entering text to a pen-based computer is described. The technique proposes a new alphabet, where each letter is a flick gesture. These flick gestures are self-disclosing using pie menus. An experiment determined the speeds of executing the flick gestures and the transition speeds between gestures. An assignment of characters to gestures is developed and evaluated. Audio feedback is used to convey whether a gesture was well- or badly-formed. A longitudinal study showed clear progress on a learning curve. The method is compared to soft keyboards, handwriting recognition systems, and unistrokes.

KEYWORDS: Stylus, text entry, pen-based computing, audio feedback.

INTRODUCTION
Computers are ever getting smaller, and the traditional keyboard is giving way to pen-based entry of text for portable applications. Existing techniques, like handwriting recognition and soft keyboards, have certain speed limitations. This paper presents *T-Cube*, a novel, faster method of entering text with a pen on small computers.

Many substitutions for the full-sized keyboard have been tried. Some manufacturers have included miniaturized keyboards on their computers. These are often adequate for the entry of phrases of text, but are inappropriate for the entry of large amounts of data. Even at their smaller sizes, they often dominate the shape and size of the small computer.

Others have attempted to use "chord" keyboards, where a small number of buttons (typically five) are pressed in combination to produce a single character. Despite the variety of chord keyboard designs, they have not succeeded in the marketplace, probably because they are difficult to learn.

Instead of a keyboard, many manufacturers have turned to a stylus as the primary mode of input. Typically, a stylus is used in two ways to enter text: handwriting recognition or soft keyboards. With handwriting recognition, the user writes on the screen using the stylus. Assuming fast and accurate recognition—not yet a reality—this technique is limited by the slow speed of printing or writing.

A soft keyboard is a graphical representation of a keyboard on the screen, activated by tapping keys with a stylus [8]. They are often used as an alternative or fallback for handwriting recognition systems. They are easy to understand, but consume much screen space and are slow to use.

A third technique for entering text with a pen, called *unistrokes*, was proposed by Goldberg and Richardson [5]. They tailored a new alphabet to the task of entering characters with a stylus. Unistrokes have several good properties: they are faster to write than Roman characters, they are easier to recognize algorithmically, and they take a small screen area to enter.

T-CUBE
This paper proposes a new alphabet and interaction style that has many of the advantages of unistrokes, and includes a faster, self-disclosing alphabet. Since a "flick" gesture is particularly fast with a pen, the alphabet constructed for T-Cube consists of flicks. A flick gesture has two easily controlled aspects: its starting point and its direction. Each flick starts in one of the nine cells arranged into a "target" shown in Figure 1a. The target is typically 0.3 to 0.7 inches in diameter. The direction of the flick can be vertical, horizontal or diagonal, specifying one of eight directions. The combination of nine starting cells with eight directions yields 72 different gestures. Each gesture represents a character, e. g. "w" or "7", or an operation, e. g. Backspace, Return or Shift.

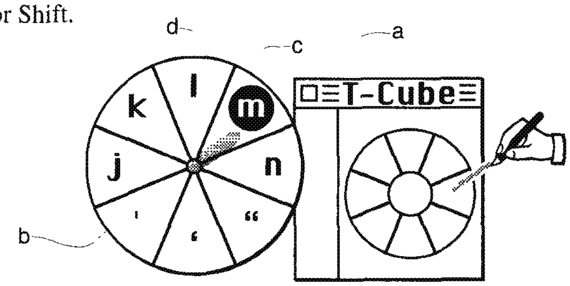

Figure 1: The T-Cube target (a) lies within a window. When the user presses the pen in one of the nine target cells, a pie menu (b) appears offset from the pen. Dragging the pen highlights one of the characters (c), and shows the result of the drag (d). When the pen is lifted, the character is "typed" to the system.

The use of starting cell and direction corresponds with the way pie menus work [2] [6] [7]. Pressing the stylus in a cell for a brief interval (one third of a second) causes a pie menu to appear (Figure 1b). The eight symbols lie in eight directions from the center of the pie. When the stylus moves a few pixels away from its starting position, the character that corresponds to that direction is highlighted (Figure 1c). The highlighted character is "typed" when the pen is released. The motion of selecting an item from a pie menu is the same as the flick gesture.

If the user remembers exactly the starting cell and direction, she may make that gesture directly. If she knows the starting cell of the desired character, she may "pop up" the pie menu to see exactly which direction the letter is in. If she doesn't remember the starting cell, she may pop up a series of pie menus until the desired symbol is found.

Until the user has learned the common characters, she needs to refer to the pop-up menus very frequently. To alleviate the tedium of popping up menus, the user may switch T-Cube into its "training mode." In this mode, the pie menu is visible whenever the pen is near the T-Cube target. When the pen is held above a cell in the target, the corresponding pie menu items are visible. She then selects a menu item by pressing the pen down, dragging in one of the eight directions, and releasing.

A typical pie menu is displayed with its center exactly under the stylus tip. With a pen-based computer, the user's hand obscures nearly half of a typical pie menu. The T-Cube pie menu is offset from the center of the target, but it is still activated by a short radial movement of the pen. The motion of the pen from its location when the pie menu popped up to its present location is represented by a thick line radiating from the center of the pie menu, as shown in Figure 1d. The pie menu is offset to the left for a right-handed user, and to the right for a left-hander.

CHARACTER LAYOUTS

An assignment of characters to gestures is called a layout. A layout is desirable if it is both predictable and fast. With a predictable layout, a moderately experienced user can guess at the gesture to make a desired character, and be right some (hopefully most) of the time. In a fast layout, frequent letters correspond to fast gestures. A predictable layout helps the novice user; a fast layout helps the expert user. To assess a layout's speed, an experiment was conducted to determine the speeds of individual gestures and transitions between gestures.

Gesture Speeds Experiment

Twenty-two subjects participated in the experiment—Apple Computer employees who were selected for their exposure to graphic tablets from a volunteer subject pool. The experiment was run on a custom pen-based computer with an untethered pen and an 8.5" (diag.) backlit LCD screen. A T-Cube target was visible on the screen. For each trial, a stimulus target appeared 0.9" above it, marked with a gesture. The subject attempted to reproduce the indicated gesture. Another trial followed after 100ms. The time from the presentation of the stimulus to the pen-down and pen-up were

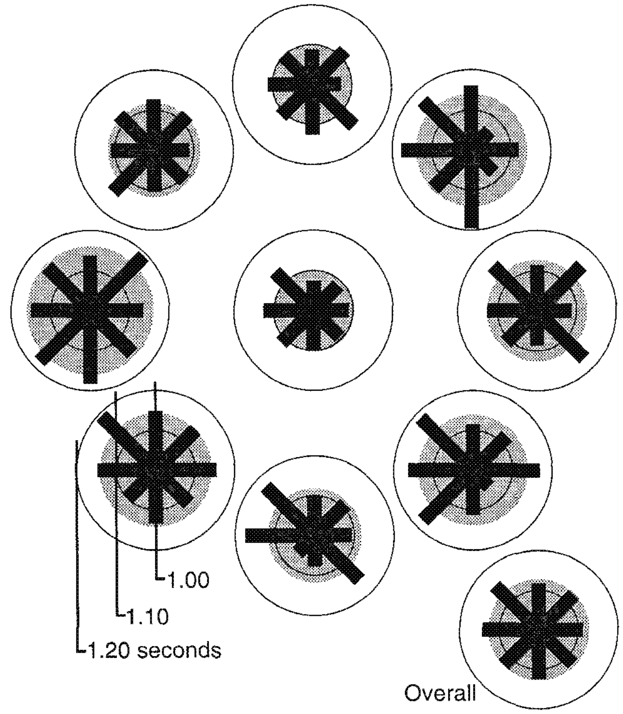

Figure 2: Mean speeds are shown for each direction and starting cell in this group of radial bar charts. The gray circle is the mean for the cell, independent of direction.

recorded, along with several other parameters. Each subject performed 72 training trails, then three blocks of about 350 trials each. In all, over 20,000 non-training trials were recorded.

This experiment resulted in two kinds of information. First, the speed for each gesture was determined. This guided the placement of frequent characters to fast gestures in the layout. Second, a matrix of gesture-to-gesture transition times was found. These times were used to evaluate the speed of layout candidates.

The results of the experiment are summarized in Figure 2. The speed value shown for each gesture is the mean of all times from stimulus to pen-up. The speeds indicated are slower than they would be in real use, in part since the subject had to look from the upper target to the lower. The fastest starting cells, in order, are Center, North, Northwest, South and East. The radial gestures (e. g. starting in the North cell and drawing to the North) are the fastest overall. Westerly gestures from any cell tend to be the slowest. There appears to be no categorical speed difference for orthogonal vs. diagonal gestures. There appears to be asymmetry in the response times, perhaps due to the handedness of the subjects.

A Predictable Layout

The construction of a predictable layout is considered next. Since the pie menu can be examined to find the direction for a character when given the correct starting cell, the assignment of characters to starting cells is more important than the arrangement within the cell's pie menu. Characters must be clustered into starting cells in a way that allows the user to

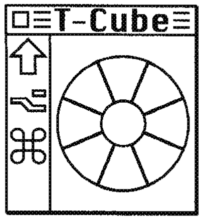

Figure 3: Symbols are displayed next to the target when modes—Shift, Option and Command—are active.

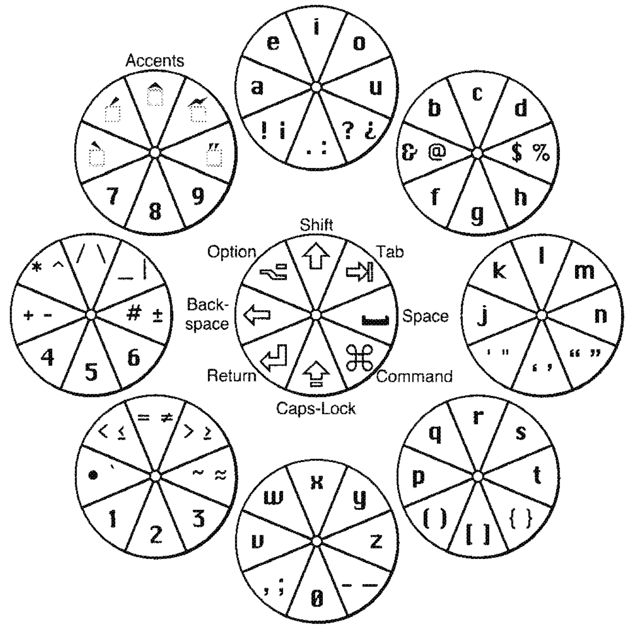

Figure 4: The character that results from each gesture is shown here. For example, the gesture that starts in the East cell and goes Northeast results in an "m". The character produced depends on the state of the Shift modifier.

predict which cell contains the desired character. Where possible, frequently used characters were associated with fast gestures.

The most frequently used characters are Space, Backspace and Shift. They are clustered with the other "non-printing" characters—Tab, Return and the so-called "meta-keys"—and placed into the fastest starting cell. For the Macintosh, meta-keys are ⌘ and Option; other systems have different ones, e. g. Alt, Ctl or ◊. Meta-keys act as toggles—the same gesture turns them on and off—and are shown next to the T-Cube target when they are active (Figure 3). Shift and Caps-Lock are meta-keys that also have an effect on the layout. As on a normal computer keyboard, when either is active, letter gestures make capital rather than lower case characters. When Shift is active, a gesture may result in a different character altogether, just as "shift-3" on a keyboard produces a "#" symbol. All the meta-keys except Caps-Lock deactivate after a character is produced. This cluster of characters and meta-keys is placed in the center starting cell (Figure 4).

Since the main goal is for the entry of text, the clustering of the letters is crucial for a predictable layout. One predictable clustering of the letters was designed by separating the vowels from the consonants:

ᴬBCDᴱFGHᴵJKLMNᴼPQRSTᵁVWXYZ

Moving the vowels into one cluster reveals a natural clustering of the consonants:

AEIOU BCD FGH JKLMN PQRST VWXYZ

The two small clusters can be combined, leaving five clusters. The clusters are placed clockwise from the North cell, as shown in Figure 4.

The ten digits must be placed in two or more cells. Choosing to mimic a numeric keypad in organization, the numbers are placed in cells along the West side of the target.

Symbols are grouped with either the numbers or the letters, depending on which they are more commonly used with. The symbol shown on the left is produced when Shift is inactive, and the right when it active.

Finally, the remaining five gestures are assigned to the Accent meta-keys (Northwest starting cell, shown in Figure 4). Accent meta-keys act as toggles, and only one can be active at a time. The active accent is shown next to the T-Cube target as below the other meta-key states. When an accent (`) is active and the user makes a letter that can use that accent ("a"), the accented character is produced ("à"), and the accent mode deactivates.

There are a myriad of possible layouts. Dozens of possible layouts were considered for this design. Future work may discover a better layout.

FEEDBACK

Any interactive system needs to give feedback to help the user understand, learn and use it. T-Cube enlists both graphic and audio feedback to help convey information about the interface. When the user makes a gesture, a small tic appears for a brief time showing an idealized version of the gesture. Many users suggested that the character itself should also appear briefly. Both are shown in Figure 5. This feedback helps strengthen the association between the gesture and the character, assisting the novice user to become an expert. It also reinforces the ideal form of the gesture—starting at the center of the cell, and extending orthogonally or diagonally.

Another way that T-Cube helps users draw well-formed gestures is with sound. A brief sound clip is heard when the user begins a flick, and another when it ends. The first sound is a subtle but crisp "tick" when the start of the flick is near the center of the cell. When it is near to the edge, a "clonk" sound is mixed in with the "click." The sound at the end of the flick is a crisp "tock" when the direction is orthogonal or diagonal. A more ambiguous angle produces the "tock" blended with a kind of "buzz" sound. The use of sound may alert users to sloppy gestures, and the potential errors that may result.

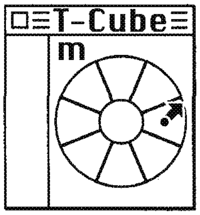

Figure 5: After a gesture is completed, the character is "typed" to the system, and graphics briefly show the character and an idealized version of the gesture.

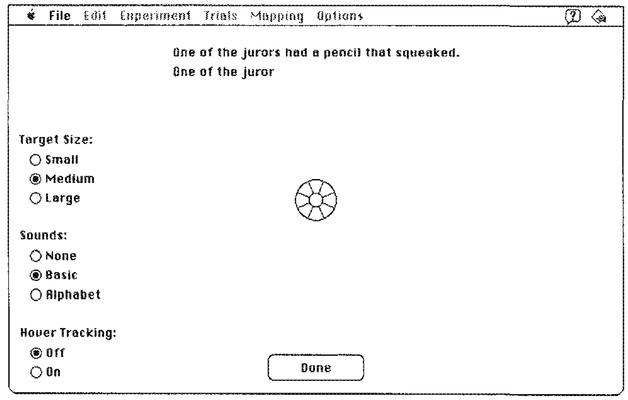

Figure 6: Subjects were studied over repeated exposure to T-Cube. In each session, they copied sentences. They had control of the target size, the sounds, and the training mode. (Screen shot shown at half actual size.)

Other sounds help convey modes. When a meta-key is invoked, a two-tone rising motif [1] is heard if it toggles on, and a two-tone descending motif plays if it toggles off. The Caps-Lock meta-key has more dramatic versions of the rising and falling motifs. The non-printing characters produce a sound since often the visual change associated with them is subtle.

The user has the option of hearing a sound for each letter written. The sound consists of the spoken letter (e. g. "w" plays a recording of the words "double you") mixed with a chime sound whose pitch varies depending on the starting cell. The upper cells produce a high pitches, the low cells produce low pitches. This is intended to reinforce the association between the cell and the letter.

Sounds in T-Cube encode much information. It is not necessary that the user perceive and use all the information all the time. Sounds may gain or lose value as the user progresses from novice to expert. What is more important is that the sounds provide a sonic texture to the user, so that changes in that texture will alert the user to changes in the interface and its use [4].

PILOT LONGITUDINAL STUDY

The design proposed above is tantalizing, but it is not clear that the resulting interface can be learned and used by real people. To determine this, a pilot longitudinal study was performed. Eleven subjects, some of whom had participated in the earlier experiment, were exposed to T-Cube several times over the course of a month. Eight subjects performed eight or nine sessions; three subjects completed only four, five or six sessions. In each session, they were presented with a series of sentences to enter using T-Cube, as shown in Figure 6. One hundred sentences were selected from the text of "Alice in Wonderland," and were presented in random order. At the end of a sentence, the subject tapped the "Done" button, and could rest for an arbitrary time before starting the next sentence. The total time spent entering characters during the session was twenty to thirty minutes.

Subjects were allowed to change at any time the target size, the kind of sounds presented, and whether they were in training mode or expert mode. The target was either 0.4, 0.55 or 0.7 inches in diameter. The interface was silent, used only the basic sounds, or used both the basic and the alphabetic sounds. At the start of each session, the medium target was presented, with basic sounds, in expert mode.

Each subject was assigned to one of two layouts. Eight of the subjects used the layout that was rated the fastest, the other three used a layout similar to the one described above.

The basic sounds were preferred by most subjects. Seven users chose basic sounds more than the other settings. Two subjects only used the sounds for a few strokes, then turned them off. Another two tried sounds both basic and off, and tended toward turning them off. While this indicates good acceptance (64%) when users have a choice of sounds in the interface, it is hard to say whether sounds made T-Cube easier to use. The alphabetic sounds were universally disliked.

The medium-sized target was preferred by most subjects. Seven subjects chose the medium target more often than the small or large targets. Two subjects preferred the large one, and two preferred the small one. Providing a variety of target sizes appears to be desirable.

Two learning curves were examined: change in speed over time, and the use of the pie menu over time. Figure 7 shows the increase in median speed for letters, space and backspace that were entered without using training mode or the pop-up pie menu, for each subject by session. For the duration of the study, the entry speed increases linearly; there is no sign of leveling off. While this indicates that the study did not run for enough sessions, it also suggests that the maximum speed observed (60 to 106 characters per minute) is significantly less than the true maximum speed for T-Cube. There is no clear speed difference between the two layouts.

Subjects' reliance on the pie menu is shown in Figure 8. In the first session, most used the training mode, while a few chose instead to use the pop-up menu. By the second session, 8% of the gestures were drawn without the aid of the menu. By the final session, subjects referred to the menu only a few times.

From observing the subjects, it was apparent that about half of the subjects were familiar with the layout after the first

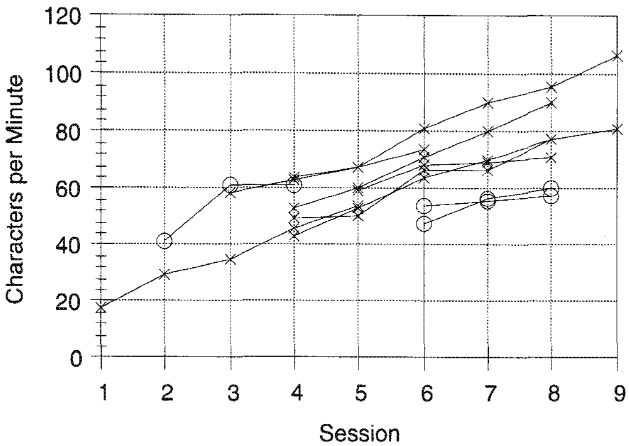

Figure 7: The entry speed increased linearly with exposure. A line, representing a subject, starts at the first session where they entered more than a few strokes by gesture only, and ends at the last session. The "x" and "o" marks indicate different layouts.

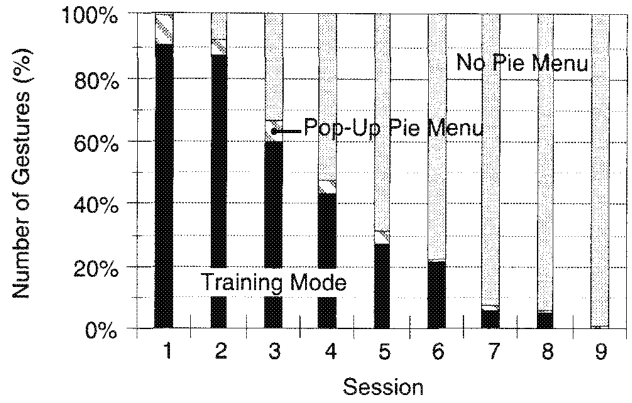

Figure 8: Almost all subjects used the training mode for the early sessions. By the end, all of them had progressed to expert mode, and were using the pop-up pie menu very little.

session, and that all subjects felt comfortable with it after the second session.

These ways of looking at the learning curve indicate that users can get to know T-Cube in a few hours of exposure. Even in that short time, good speeds can be achieved. Given longer to learn, greater speeds could be achieved.

Subjects consistently reported a few observations. Parallax caused by the distance between the front glass and the LCD image made users uncomfortable. Some mentioned that they write at an angle, so North should be customizable, rather than fixed to be "up" on the screen. Many suggested that separate buttons be added for the most common functions, like backspace, shift and repeat.

COMPARISON

How does T-Cube compare to the other ways of entering text with a stylus? Some of the many factors are considered here:

Screen Size: T-Cube uses less than a square inch of screen space. Unistrokes take somewhat more. Soft keyboards take vastly more space. Handwriting can be done either in-place or in a separate entry window. When it is done in-place, it takes up virtually no space. When entry is done into a special area, that area can be very large.

Speed: The speeds of T-Cube, unistrokes [5], finger-activated soft keyboards [8], writing with pen and paper, and typing [3] are compared in Figure 9. T-Cube rates competitively against the others. Note that the speed range for unistrokes is based on the "peak error-free writing speeds" for two users, so may not represent the population mean.

Character Set: Handwriting systems and unistrokes have severely limited character sets. T-Cube produces all of the ASCII characters, along with many of the extensions that are common and useful in today's international systems. Soft keyboards can produce any character that a real keyboard can.

Visual Attention: Whether an input modality requires visual attention impacts how and when it can be used. One that doesn't require the eyes can be used for transcription, while driving a car, or while talking to a client. One of unistrokes' greatest strength is its ability to be used without visual attention. Handwriting and soft keyboards both require a great deal of visual attention.

Using T-Cube, the entire target is accessible without moving the hand. It is conceivable that T-Cube could be used without looking at it, relying on the user's proprioception to guide the pen back to the center of the target. The difficult part for a user would be accurately getting to the desired starting cell. Audio feedback that acted as a homing device to find the cell centers might help to locate them.

Further experiments are needed to determine whether T-Cube can be used with little or no visual attention. A redesign of T-Cube employing hierarchical marking menus [6] could much better serve the "eyes-free" need.

Novice and Expert Use: While handwriting recognition systems appear to be novice-oriented, they are often difficult for the novice until he develops a sense of what the recognizer thinks is a well-formed character; once he learns this, the accuracy rate will climb. As he gains experience though, there is no way that he can enter text faster than his handwriting speed will allow. Handwriting systems have poor novice performance and no path for expert growth.

Soft keyboards are very accessible to the novice. They use a layout that is likely to be known to the user. The expert is limited, though, by the speed of tapping buttons on the screen.

The novice user of unistrokes must use a reference card initially, though it is unclear for how long. Once the user does successfully become an expert, great speed gains can be made. Writing speed after one week's experience compares favorably with soft keyboards. The reference card makes unistrokes cumbersome for novices, but the potential speed gains makes it good for experts.

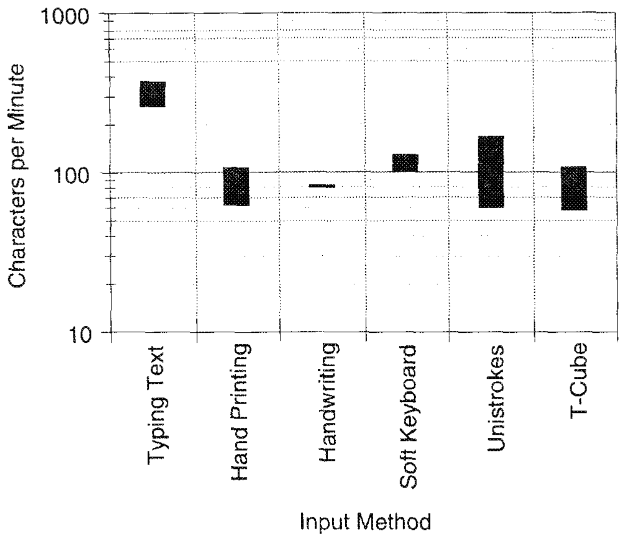

Figure 9: Entry speeds for various methods are compared on a logarithmic scale. After eight or nine sessions, T-Cube subjects attained speeds in the indicated range. Since all of the subjects were continuing to get faster at the end of the observation, it is expected that in actual use, speeds would be higher than indicated.

T-Cube supports both the novice and the expert user. The novice can use the predictable layout of letters and the pie menu to find characters. The act of using the interface in this way helps the user to gain expertise. For the expert user, entry of text becomes a fairly simple motor task, with potential for high speed.

Best Use

By examining the strengths of T-Cube relative to the other techniques, a picture of its use can be painted. It requires a small screen area, and is fast for experts. It helps the novice to become an expert—the training process is built into the interface. These characteristics paint a picture of a user of a portable, hand-held device who, motivated by the payoff of fast entry of text, is willing and able to invest the time to become an expert. This device might be a note-taking computer or a interactive TV remote control. On the other hand, T-Cube is not appropriate for many situations, as when use is infrequent (e. g. an information kiosk) or when a keyboard is available (e. g. a laptop or desktop computer). It is unclear whether any of the pen-based techniques are appropriate for entering large amounts of data.

CONCLUSION

T-Cube proposes a new alphabet for pen-based computing that is more appropriate for this medium than the traditional "paper" alphabet. In addition, it shows how a certain class of gestures can be self-disclosing, using pie menus.

T-Cube supports the novice user in a variety of ways. Pie menus reveal how a gesture is formed, as well as the association between characters and gestures. A predictable layout of letters, numbers and punctuation helps the novice guess which starting cell to choose. The training mode, in which the pie menu is always visible, showing the contents of the starting cell that is under the pen tip, further helps the novice search for characters in the layout. When the user gains experience, she can leave the training mode, but still refer to the pie menu when the need arises, by pressing in the starting cell. The pie menu reveals the direction to draw the gesture to make the desired character.

As the novice uses T-Cube, she is tacitly training herself to become an expert. The motion that she uses to invoke a pie menu is exactly the flick gesture that an expert uses to create a character. Frequent entry of a common character leads to a strong association with the specific gesture that stands for character. As she gains this knowledge, entry speed increases dramatically. When she becomes an expert, she is rewarded with a fast, predictable, reliable way to enter text.

Future research may examine the issues of using T-Cube for the young, the elderly, or the disabled. Also of great interest is the application of these techniques to other languages, particularly to Asian character sets with the use of a deeper hierarchy of pie menus.

REFERENCES

1. Blattner, M., Sumikawa, D., and Greenberg, R. Earcons and icons: Their structure and common design principles, in *Human-Computer Interaction 4*, 1 (1989), pp. 11-44.

2. Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. An empirical comparison of pie vs. linear menus, in *Proc. CHI '88* (Washington, D. C., May 15–19, 1988), ACM Press, pp. 95-100.

3. Card, S., Moran, T., and Newell, A. *The psychology of human-computer interaction*. Erlbaum, Hillsdale, New Jersey, 1983, p. 61.

4. Gaver, W., Smith, R., and O'Shea, T. Effective Sounds in complex systems: The ARKola simulation, in *Proc. CHI '91* (New Orleans, April 28–May 2, 1991), ACM Press, pp. 85-90.

5. Goldberg, D., and Richardson, C. Touch-typing with a stylus, in *Proc. INTERCHI '93* (Amsterdam, April 24–29, 1993), ACM Press, pp. 80-87.

6. Kurtenbach, G., and Buxton, W. The limits of expert performance using hierarchic marking menus, in *Proc. INTERCHI '93* (Amsterdam, April 24–29, 1993), ACM Press, pp. 482-487.

7. Ludolph, F., Chow, Y., Ingalls, D., Wallace, S., and Doyle, K. The Fabrik programming environment, in *1988 IEEE Workshop on Visual Languages,* pp. 222-230.

8. Sears, A., Revis, D., Swatski, J., Crittenden, R., and Shneiderman, B. Investigating touchscreen typing: The effect of keyboard size on typing speed. *Behaviour & Information Technology 12*, 1 (1993), pp. 17-22.